

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-24958

(43) 公開日 平成11年(1999) 1月29日

(51) Int.Cl.⁶

G 0 6 F 11/28

識別記号

3 1 0

F I

G 0 6 F 11/28

3 1 0 A

審査請求 有 請求項の数10 F D (全 23 頁)

(21) 出願番号 特願平9-189322

(22) 出願日 平成9年(1997) 6月30日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 米澤 直道

東京都港区芝五丁目7番1号 日本電気株式会社内

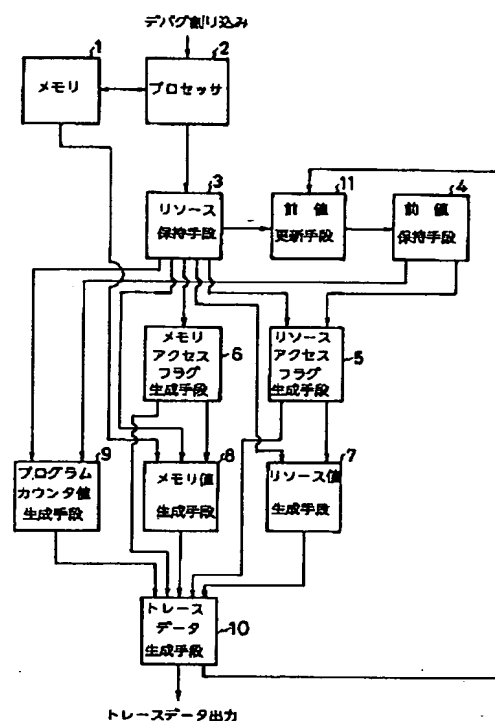
(74) 代理人 弁理士 河原 純一

(54) 【発明の名称】 トレースデータ削減方式

(57) 【要約】

【課題】 情報処理装置の各リソースの持つ特性に注目したトレースデータの生成を行うことにより、トレースデータの情報量の削減を図る。

【解決手段】 リソースアクセスフラグ生成手段5はリソース保持手段3のリソースの値と前値保持手段4のリソースの値とを比較し、アクセスされたリソースと1対1に対応したビットパターンを生成する。リソース値生成手段7はビットパターンに基づいてリソース保持手段3からアクセスされたリソースの値を出力する。トレースデータ生成手段10はビットパターンとリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータ出力すると同時にトレースデータ出力完了指示を出力し、前値更新手段11はこの指示によりリソース保持手段3のリソースの値で前値保持手段4のリソースの値を更新する。



【特許請求の範囲】

【請求項1】 1命令毎にデバッグ割り込みを発生させることが可能であり、かつデバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段を有する情報処理装置において、

1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段と、

前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段と、

このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段と、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段と、

このトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段とを有することを特徴とするトレースデータ削減方式。

【請求項2】 命令語の中に、アクセスするメモリのアドレスを指定するアドレス指定フィールドを1つもしくは複数有する情報処理装置において、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアccessフラグ生成手段と、このメモリアccessフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段とを有するとともに、事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力する前記トレースデータ生成手段を有する請求項1記載のトレースデータ削減方式。

【請求項3】 前記リソースビットパターンの中のプログラムカウンタに対応するビットの値を、トレースデ

ータ内にプログラムカウンタ全体の値を保持するか、または1命令前のプログラムカウンタの値との差分のみを保持するかを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成する前記トレースデータ生成手段を有する請求項1または請求項2記載のトレースデータ削減方式。

【請求項4】 前記リソースビットパターンの中のプログラムカウンタに対応するビットを1ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が1命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウ

10

20

30

40

50

3

ンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成する前記トレースデータ生成手段を有する請求項1または請求項2記載のトレースデータ削減方式。

【請求項5】 コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項6】 コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアccessフラグ生成手段、このメモリアccessフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアド

4

レスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段、事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンとを組み合わせたりソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項7】 コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または1命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了

5

指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項8】 コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアccessフラグ生成手段、このメモリアccessフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段、前記リソースビットパターンの中のプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または1命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時

6

にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項9】 コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、前記リソースビットパターンの中のプログラムカウンタに対応するビットを1ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が1命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値を含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ

7

生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項10】 コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアccessフラグ生成手段、このメモリアccessフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段、前記リソースビットパターンのうちのプログラムカウンタに対応するビットを1ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が1命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカ

8

ウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はトレースデータ削減方式に関し、特に情報処理装置の性能を解析するために実際の情報処理装置上でユーザプログラムがどのようなシーケンスで実行されるのかを表す実行履歴であるトレースデータを削減して記録するトレースデータ削減方式に関する。

【0002】

【従来の技術】従来、性能評価のために命令の実行履歴であるトレースデータを採取する場合、1命令毎にデバッグ割り込みを発生させ、このデバッグ割り込みによりスタックフレーム上などに格納された情報処理装置の持つプログラムカウンタ、命令語、レジスタなどのリソースの値を、そのまま外部記憶装置に出力していた。そのため、一般的な情報処理装置が1つの命令でアクセスできるリソースの数が1から数個程度であるのに対し、外部記憶装置には更新されていないリソースの値を含むすべてのリソースの値が記録されることになり、効率の悪いデータ列となっていた。

【0003】そこで、1命令毎に変化したリソースの値だけを記録することにより、外部記憶装置に記憶されるデータの効率を改善することが可能であり、その従来技術として、特開平7-191875号公報（以下、先行技術文献という）に開示されたデバッガがある。この従来のデバッガには、プログラムを1ステップずつ実行するとともに、1命令実行する毎に情報処理装置の状態を出力するトレース実行手段と、トレース実行手段の出力する情報処理装置の状態を保持する実行状態保持手段と、トレース実行手段の出力する情報処理装置の状態と実行状態保持手段の情報とから変更履歴を作成する実行出力抽出手段と、この変更履歴を記憶する変更履歴保持手段とが含まれている。ただし、上記先行技術文献には、変更履歴の構成や構造に関する記述はない。また、先行技術文献第3頁右段の25行目から29行目に示される実施例の記述から、変更のあったメモリやレジスタの種類と、レジスタ番号やメモリのアドレスおよび実際

の値からなる履歴情報であると考えられる。加えて、上記先行技術文献において、実行出力抽出手段は、変更のあった情報を出力する以外、特段の機能を有していないことから、これら履歴情報は、履歴情報間の相互関係などを意識した構造になっていないと考えられる。また、変更された情報列のみを変更履歴保持手段に記憶した場合、1命令分の区切りを認識することが不可能となってしまうため、必然的に1命令分を識別するための区切り情報が必要になると考えられる。

【0004】以上から、先行技術文献の変更履歴保持手段に記録される1つのリソースに関する履歴情報は、図12に示すような形式になると考えられる。

【0005】図12は、1つの履歴情報に関するデータ形式の例であり、ここでは、これを単位情報と呼ぶ。この単位情報には、そこに保持されている情報の種類、たとえばプログラムカウンタ、レジスタ、メモリといったリソースを示す識別情報21と、各種類毎の位置、たとえばレジスタの場合はレジスタ番号、メモリの場合はメモリのアドレスを示す位置情報22と、情報の内容そのものを示す内容23とが含まれる。

【0006】図13は、ある命令でレジスタの1番が1234の内容でアクセスされた場合のレジスタ1に関する単位情報の例を示す。図13において、識別情報21にはレジスタを示す識別情報Rが登録される。また、位置情報22にはレジスタ番号の1に対応する情報が登録される。同様に、内容23には、アクセスされたレジスタの1番の内容である1234が登録されることになる。

【0007】上記先行技術文献では、1命令毎に変化のあったリソースの単位情報を実行履歴として変更履歴保持手段に残すことができるため、この実行履歴を外部記憶装置などに出力することにより、プログラムの実行履歴であるトレースデータを作成できる。

【0008】そこで、上記先行技術文献のデバッガを使用して、トレースデータを採取した場合、どのようなトレースデータが採取されるかについて説明する。

【0009】まず、図2はトレースデータを採取する情報処理装置の条件、図14はリソースと単位情報の中の識別情報との対応表を示す。

【0010】図5は、トレースデータを採取する5つの命令の実行イメージを1命令毎に別々の表で示している。図5の表の見方は、リソースの列がアクセスされるリソースの種類、位置の列が各リソースの位置情報、フィールド位置が命令語内のアドレス指定フィールドの位置、アクセス内容がアクセスされた各リソースの値を示す10進の値である。

【0011】図5において、位置の欄が“-”となっているものは、その欄に対応するリソースが1つしかなく、位置情報を必要としないものである。フィールド位置の欄が“-”となっているものは、その欄に対応して

いるリソースがメモリではなく、アドレス指定フィールドを必要としないものである。

【0012】たとえば、図5の1命令目の表は、命令語が10、プログラムカウンタが1002、レジスタ2およびレジスタ8がそれぞれ444および666の値でアクセスされ、命令語のアドレス指定フィールドの1番で示された120番地のアドレスのメモリの888の値でアクセスされたことを示している。

【0013】図15は、図2および図14で示した条件により、図5で示した5つの命令を順に実行した場合にトレースデータとして採取される実行履歴を示した表である。

【0014】図15において、トレースデータと記述された列の、識別情報、位置情報および内容がトレースデータとして出力される値である。まず、図5に示した1命令目が実行されると、はじめに命令の区切りを示すための識別情報である“S”を持ったトレースデータが出力される。次に、1命令目ではプログラムカウンタおよび命令語が内容1002および10でアクセスされていることから、識別情報および内容が“P”および1002ならびに“1”および10となるトレースデータが出力される。また、1命令目ではレジスタ2、8およびメモリの120番地がそれぞれ内容444、666および888でアクセスされていることから、それぞれの識別情報、位置情報および内容を“R”，2および444と、“R”，8および666と、“M”，120および888としたトレースデータが出力され、これが図15の1命令目の行に記述されたトレースデータに対応する。

【0015】次に、図15の2命令目が実行されると、まず、1命令目と同様に、命令の区切りを示すために区切り情報“S”を持った識別情報が出力される。次に、2命令目では、プログラムカウンタおよび命令語が1006および456の内容でアクセスされていることから、識別情報および内容を“P”および1006ならびに“1”および456としたトレースデータが出力される。また、2命令目では、レジスタ15およびメモリ1の480番地が135および1010の内容でアクセスされていることから、それぞれの識別情報、位置情報および内容を“R”，15および135ならびに“M”，480および1010としたトレースデータが出力され、これが図15の2命令目の行にかかれたトレースデータに対応する。

【0016】同様に、図5の3、4および5命令目を実行した場合に後で採取されるトレースデータを示したのが、図15の3、4および5命令目の行である。

【0017】以上から、図15のトレースデータの列が、図5の1命令目から5命令目までを順番に実行した場合のトレースデータに対応することになる。

【0018】ここで、この従来例におけるトレースデー

タの情報量について検討する。まず、トレースデータ上の各情報の情報量を、図7に示す。図7において、識別情報のビット数は、図14の表から少なくとも5種類必要であることから3ビットとしている。命令語長は、図5に示されている命令語長が対応する。

【0019】図7のトレースデータ上の情報量の条件により、従来例で採取したトレースデータの情報量を示したのが、図15中の情報量の欄であり、各欄の値は対応する単位情報毎のビット数を示す。また、合計の行に、5命令分すべてのトレースデータの情報量を示す。したがって、この従来例では、図2および図14に示した条件下において図5で示した5つの命令列を実行した場合、採取されるトレースデータのビット数は627ビットとなる。

【0020】以上のように、上記先行技術文献のデバッガーにおいても、命令毎に1命令前との変更部分をチェックし、変更された情報のみを記憶していくことで、すべてのリソースの値を保持することなく各命令毎の実行履歴であるトレースデータを採取することが可能である。

【0021】しかしながら、情報処理装置の持つ情報にはいくつかの特性がある。たとえば、通常、情報処理装置では、1命令毎にプログラムカウンタ、命令語、レジスタなどの複数のリソースがアクセスされる。したがって、従来例のように更新されたリソースを識別するため、1つのリソース毎に専用の識別情報を持たせると、トレースデータ上では、アクセスされた情報の数に比例した識別情報が必要になる。加えて、トレースデータへの出力が必要となるレジスタなどのすべての数は、それぞれの情報処理装置のアーキテクチャや性能解析を行いたい内容にも依存するが、数十程度の場合が一般であり、その場合、リソース毎に識別情報を与えるより、各リソースと1対1で対応したビットパターンで表現した方が、同時に複数のリソースに対するアクセスの有無を表現できることから、トレースデータ内に登録されるリソースの数に比例して増える識別情報が不要な分、全体としての情報量を減らすことが可能である。

【0022】また、メモリのアクセスについては、一般の情報処理装置の場合、命令語の値と命令実行時のレジスタの値とが既知であれば、その命令によってアクセスされるメモリのアドレスを導き出すことが可能であり、上記先行技術文献のように、メモリの位置を実行履歴として持つ必要はなく、無駄な情報である。

【0023】加えて、プログラムカウンタは、通常、1命令毎に更新されるとともに、多くの場合、実行された命令の命令語長でインクリメントされるため、1命令毎にプログラムカウンタ全体の値を実行履歴として記録することには無駄が多い。

【0024】以上の点から、上記先行技術文献のように、1命令前のリソースの値と現在のリソースの値との

差分から単純に実行履歴を作成する方法を流用することでも、すべてのリソースの値を保存する場合に比べ、データ量を削減したトレースデータを作成することは可能である。しかし、情報処理装置の特性を考えた場合、無駄な情報が多く含まれる情報となってしまう、数億命令単位の実ジョブのトレースデータを採取するためには、上記先行技術文献の構成を利用したトレースデータの採取方法は機能的に不十分であった。

【0025】

10 【発明が解決しようとする課題】第1の問題点は、上記先行技術文献に記載されたデバッガーでは、情報処理装置の状態を示す情報の持つ特性を意識せず、単純に1命令前のリソースの値と現在のリソースの値との変更部分をその他の情報に頼ることなく単独の情報で再生できるように、リソースの値とその情報の位置やリソースの種類などを識別する情報を履歴情報として記録している点にある。その理由は、通常の情報処理装置では、1命令毎にプログラムカウンタ、命令語、レジスタなどの複数のリソースがアクセスされるため、1つのリソース毎に識別情報を付与すると、トレースデータ上ではアクセスされたリソースの数に比例して、リソースを識別する情報も増えるからである。

【0026】また、プログラムカウンタは通常1命令毎に実行した命令語長分でインクリメントされるだけであり、プログラムカウンタが変更されても、多くの場合命令語長がわかれば、1命令前のプログラムカウンタの値に命令語長を加えることにより、新しいプログラムカウンタの値を求めることが可能であることから、プログラムカウンタの値の全てを記憶しておく必要はない。

30 【0027】加えて、メモリのアクセスするアドレスは、実行された命令語の値と、その命令実行時のレジスタの値とが既知であれば、それらの値から計算することが可能であり、変更されたメモリの位置であるアドレスを記憶しておくことは無駄である。

【0028】本発明の目的は、ユーザプログラムの性能評価に使われる命令の実行履歴であるトレースデータの採取において、1命令前のプログラムカウンタやレジスタなどのリソースの値と現在のリソースの値との差分によってトレースデータを生成する以外に、情報処理装置の各リソースの持つ特性に注目したトレースデータの生成を行うことにより、トレースデータの情報量を削減するようにしたトレースデータ削減方式を提供することにある。

【0029】また、本発明の他の目的は、1つの命令で多くのリソースがアクセスされ、それらリソースの値がトレースデータとして出力された場合でも、1命令分のトレースデータの中にどのリソースに対応する情報が登録されているかの識別が、1つのビットパターンのみで実現できるようにし、トレースデータに出力されるリソースの数に比例して増加するリソースを識別するための

情報を排除できるようにしたトレースデータ削減方式を提供することにある。

【0030】さらに、本発明の別の目的は、メモリの内容が更新された場合、命令語のアドレス指定フィールドに対応したビットの値と、アクセスされたメモリの内容だけがトレースデータに出力されるようにし、メモリの位置を示すアドレスがトレースデータ上に出力されないようにしたトレースデータ削減方式を提供することにある。

【0031】さらにまた、本発明のさらに別の目的は、プログラムカウンタに関しては、多くの場合、トレースデータ上に出力されるプログラムカウンタに対応した情報は、トレースデータの中にプログラムカウンタ全体の値があるのか、1命令前の値との差分のみがあるのかを示すビットの情報および命令語長、または命令語長の情報だけとなるようにし、1命令毎に常にプログラムカウンタの値がトレースデータに出力される場合に比べ、トレースデータの情報量を大幅に減らすことができるようにしたトレースデータ削減方式を提供することにある。

【0032】

【課題を解決するための手段】本発明のトレースデータ削減方式は、1命令毎にデバッグ割り込みを発生させることが可能であり、かつデバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段を有する情報処理装置において、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段と、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段と、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段と、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段と、このトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段とを有する。これにより、たとえば、1つの命令で多くのリソースがアクセスされ、それらリソースの値がトレースデータとして出力された場合でも、1命令分のトレースデータの中にどのリソースに対応する情報が登録されているかの識別が、1つのビットパターンのみで実現できるため、トレースデータに出力されるリソースの数に比例

して増加するリソースを識別するための情報を排除できる。

【0033】また、本発明のトレースデータ削減方式は、請求項1記載のトレースデータ削減方式において、命令語の中に、アクセスするメモリのアドレスを指定するアドレス指定フィールドを1つもしくは複数有する情報処理装置において、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアccessフラグ生成手段と、このメモリアccessフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段とを有するとともに、事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力する前記トレースデータ生成手段を有する。これにより、メモリの内容が更新された場合、命令語のアドレス指定フィールドに対応したビットの値と、アクセスされたメモリの値だけがトレースデータに出力されるため、メモリの位置を示すアドレスがトレースデータ上に出力されない。

【0034】さらに、本発明のトレースデータ削減方式は、請求項1または請求項2記載のトレースデータ削減方式において、前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または1命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の

出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成する前記トレースデータ生成手段を有する。これにより、プログラムカウンタに関しては、多くの場合、トレースデータ上に出力されるプログラムカウンタに対応した情報は、トレースデータの中にプログラムカウンタ全体の値があるのか、1命令前のプログラムカウンタの値との差分のみがあるのかを示すビットの情報および命令語長、または命令語長の情報だけとなるため、1命令毎に常にプログラムカウンタの値がトレースデータに出力される場合に比べ、トレースデータの情報量を大幅に減らすことができる。

【0035】さらにまた、本発明のトレースデータ削減方式は、請求項1または請求項2記載のトレースデータ削減方式において、前記リソースビットパターンの中のプログラムカウンタに対応するビットを1ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が1命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成する前記トレースデータ生成手段を有する。

【0036】一方、本発明の記録媒体は、コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令

語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【0037】また、本発明の記録媒体は、コンピュータを、デバッグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバッグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアクセスフラグ生成手段、このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段、事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指

示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【0038】さらに、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または1命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【0039】さらにまた、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保持手段、1命令前のデバグ割り込み発生時のリ

ソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアccessフラグ生成手段、このメモリアccessフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段、前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または1命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアccessフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と前記メモリ値生成手段の出力するメモリの値と前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【0040】また、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース保

持手段、1命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、前記リソースビットパターンの中のプログラムカウンタに対応するビットを1ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が1命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【0041】さらに、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ、命令語、レジスタなどのリソースの値を保持するリソース

保持手段、1命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと1対1に対応したビットパターンを生成するリソースアクセスフラグ生成手段、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと1対1に対応したビットパターンを生成するメモリアクセスフラグ生成手段、このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段、前記リソースビットパターンの中のプログラムカウンタに対応するビットを1ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が1命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手

段、ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【0042】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照しながら詳細に説明する。

【0043】図1は、本発明の第1の実施の形態に係るトレースデータ削減方式の構成を示すブロック図である。本実施の形態に係るトレースデータ削減方式は、メモリ1と、プロセッサ2と、リソース保持手段3と、前値保持手段4と、リソースアクセスフラグ生成手段5と、メモリアクセスフラグ生成手段6と、リソース値生成手段7と、メモリ値生成手段8と、プログラムカウンタ値生成手段9と、トレースデータ生成手段10と、前値更新手段11とから構成されている。

【0044】メモリ1は、プログラムやデータを保持する。

【0045】プロセッサ2は、メモリ1からプログラムやデータを受け取り処理を行う。また、プロセッサ2は、デバッグ割り込み指示に応じて、1命令毎に自分の持つ命令語、プログラムカウンタ、レジスタなどのリソースの値を出力する。

【0046】リソース保持手段3は、プロセッサ2から出力されたリソースの値により保持しているリソースの値を更新するとともに、リソース更新指示を出力する。

【0047】前値保持手段4は、1命令前のデバッグ割り込み発生時にリソース保持手段3が保持していたリソースの値を保持している。

【0048】リソースアクセスフラグ生成手段5は、リソース保持手段3からのリソース更新指示を受け取ると、リソース保持手段3の保持するリソースの値と前値保持手段4の保持するリソースの値とを比較し、リソース保持手段3の保持するリソースの値と前値保持手段4の保持するリソースの値とが一致する場合には0、リソース保持手段3の保持するリソースの値と前値保持手段4の保持するリソースの値とが一致しない場合には1となる、各リソースと1対1に対応したビットパターンを出力する。

【0049】メモリアクセスフラグ生成手段6は、リソース保持手段3からのリソース更新指示を受け取ると、リソース保持手段3の保持する命令語の値からメモリ1に対してアクセスが発生したか否かを判定すると同時に、メモリ1へのアクセスがあった場合には1、メモリ1へのアクセスがなかった場合には0となる命令語のアドレス指定フィールドに対応したビットパターンを出力する。

【0050】リソース値生成手段7は、リソースアクセスフラグ生成手段5の出力するビットパターンに基づい

てリソース保持手段3の保持するリソースの値からアクセスされたリソースの値のみを選択して出力する。

【0051】メモリ値生成手段8は、メモリアクセスフラグ生成手段6の出力するビットパターンとリソース保持手段3の保持する命令語やレジスタなどのリソースの値とから、アクセスされたメモリ1のアドレスを計算するとともに、メモリ1からアクセスされたアドレスの内容を読み出して出力する。

【0052】プログラムカウンタ値生成手段9は、リソース保持手段3からリソースアクセス指示を受け取ると、リソース保持手段3の保持するプログラムカウンタの値と前値保持手段4の保持するプログラムカウンタの値との差分から事前に指定された変位で更新されたか否かを判断し、指定された変位で更新された場合は、その変位に対応したビットパターンを出力し、指定された変位以外で更新された場合は、対応する変位を持たないビットパターンを出力するとともに、リソース保持手段3の保持するプログラムカウンタの値を出力する。

【0053】トレースデータ生成手段10は、リソースアクセスフラグ生成手段5の出力するビットパターンと、メモリアクセスフラグ生成手段6の出力するビットパターンと、プログラムカウンタ値生成手段9の出力する変位のビットパターンとを事前に設定された規則に従って再構成して出力するとともに、この出力に続き、リソース値生成手段7の出力するリソースの値と、メモリ値生成手段8の出力するリソースの値とを出力する。また、プログラムカウンタ値生成手段9の出力する変位のビットパターンが対応する変位の値を持たないパターンの場合は、トレースデータ生成手段10は、プログラムカウンタ値生成手段9の出力するプログラムカウンタの値を事前に指定された規則に従って出力する。最後に、トレースデータ生成手段10は、すべてのトレースデータの出力を完了すると、トレースデータ出力完了指示を出力する。

【0054】前値更新手段11は、次の命令のトレースデータを採取する準備として、トレースデータ生成手段10から出力されるトレースデータ出力完了指示を受け取ると、リソース保持手段3の保持するリソースの値を読み出し、読み出したリソースの値で前値保持手段4の保持するリソースの値を更新する。

【0055】次に、このように構成された第1の実施の形態に係るトレースデータ削減方式の動作について説明する。

【0056】まず、情報処理装置の前提条件については、従来例でも使用した図2の条件を使用する。

【0057】次に、リソースに対するアクセスがあったか否かを示すビットとプログラムカウンタの変位とからなるビットパターンの形式を、図3のビットパターンフォーマットとする。図3のビットパターンフォーマットにおいて、ビットR00～R15は、それぞれ16本あ

るレジスタ0～15と1対1で対応する。ビットIRは、命令語に対応する。ビットAS1およびAS2は、命令語の持つ2つのアドレス指定フィールドと1対1で対応する。ビットIL2～IL0はプログラムカウンタの変位を示すビットであり、プログラムカウンタの変位の1、2または4の値がそのまま2進値として設定されるものとする。また、ビットIL2～IL0の値がすべて1のとき、トレースデータ上にプログラムカウンタ全体の値が登録されていることを示すこととする。ここで、図4は、図3で定義したビットパターンに対する設定の例であり、レジスタ2、8、命令語および命令語のアドレス指定フィールド1で指定されたアドレスがアクセスされ、プログラムカウンタは変位2で更新された場合を示したものである。

【0058】トレースデータへの出力順は、はじめに図3のビットパターンフォーマットにしたがって作成されたビットパターンが出力され、次にリソースの値が図3のビットパターンフォーマットの定義の左から順番に出力されるものとする。たとえば、レジスタ1および2の値が出力される場合は、ビットパターン、レジスタ1およびレジスタ2の内容の順でトレースデータに出力される。

【0059】プロセッサ2は、メモリ1にあるプログラムやデータに従いプログラムを実行するとともに、デバッグ割り込みが指示された場合、指定されたデバッグ割り込み条件に従って、デバッグ割り込みを発生させ、デバッグ割り込み指示を出力すると同時に、そのときのプロセッサ2の持つプログラムカウンタや16本のレジスタ0～15などのリソースの値をリソース保持手段3に出力する。

【0060】リソース保持手段3は、プロセッサ2からデバッグ割り込みによるリソースの値を受け取ると、事前に指定された領域に各リソースの値を格納する。また、このとき、前値保持手段4には、1命令前にリソース保持手段3に保持されていたリソースの値が保持されている。リソース保持手段3は、プロセッサ2からデバッグ割り込み信号を受け取ると、まず、リソースアクセスフラグ生成手段5に対し、図3に示したビットパターンフォーマットのうち、メモリ1のアクセスがあったか否かを示すビットAS1およびAS2ならびにプログラムカウンタの変位を示すビットIL0～IL2を除くビットR00～R15およびIRからなるビットパターンの生成を指示する。

【0061】リソースアクセスフラグ生成手段5は、リソース保持手段3からのビットパターン生成指示を受け取ると、リソース保持手段3の保持する命令語および16本のレジスタ0～15の値と前値保持手段4の保持する命令語および16本のレジスタ0～15の値とを比較し、値に変化のあったリソースを調べる。そして、値に変化のあったリソースに対応するビットを1、値に変化

のなかったリソースに対応するビットを0としたビットパターンを生成する。たとえば、図5の1命令目の場合、レジスタ2、8および命令語がアクセスされていることから、図3に示すビットパターンフォーマットのうち、ビットR02、R08およびIRが1、ビットR00、R01、R03、R07、R09およびR15が0となるビットパターンを生成し、これをリソース値生成手段7に出力する。

【0062】これと平行して、リソース保持手段3は、プロセッサ2からデバッグ割り込み指示を受け取ると、メモリアccessフラグ生成手段6に対してメモリ1に対するアクセスがあったか否かを示すビットAS1およびAS2に対応するビットパターンの生成を指示する。

【0063】メモリアccessフラグ生成手段6は、リソース保持手段3からビットパターン生成指示を受け取ると、リソース保持手段3の保持する命令語の値を調べ、メモリアccessがあったか否か、またメモリアccessがあった場合には命令語のメモリ指定フィールドのいずれで指定されたアドレスかを調べる。そして、メモリ1に対するアクセスがあった場合は、メモリアccessのあった領域を指定しているメモリ指定フィールドに対応するビットを1、メモリアccessがなかったメモリ指定フィールドに対応するビットを0とした、図3のビットパターンフォーマットの中のビットAS1およびAS2に対応したビットパターンを生成し、これをメモリ値生成手段8に出力する。たとえば、図5の1命令目の場合、アドレス指定フィールド1で指定されたアドレスにアクセスがあることから、図3に示すビットパターンフォーマットのうち、ビットAS1が1、ビットAS2が0となるビットパターンをメモリ値生成手段8に出力する。

【0064】同時に、リソース保持手段3は、プロセッサ2からデバッグ割り込み指示を受け取ると、プログラムカウンタ値生成手段9に対してプログラムカウンタの変位を示すビットIL2～IL0に対応したビットパターンの生成を指示する。

【0065】プログラムカウンタ値生成手段9は、リソース保持手段3からのビットパターン生成指示を受け取ると、リソース保持手段3の保持するプログラムカウンタの値と前値保持手段4の保持する1命令前のプログラムカウンタの値とを比較し、その変位に対応するビットパターンを生成し、トレースデータ生成手段10に出力する。また、プログラムカウンタ値生成手段9は、求めた変位に対応するビットパターンがない場合は、どの変位とも対応しないビットパターンとともに、プログラムカウンタ全体の値も併せて、トレースデータ生成手段10に出力する。たとえば、図5の1命令目の場合、命令語長の2でプログラムカウンタが更新されるため、図3に示すビットパターンフォーマットのビットIL2～IL0には、2進値の2に対応する0、1および0となるビットパターンが生成され、これがトレースデータ生成

手段10に出力される。また、図5の5命令目の場合、プログラムカウンタの値が1014から2000に更新されているため、事前に規定された変位の1, 2または4のどれとも対応がとれないことから、ビットIL2~IL0がすべて1となるビットパターンがトレースデータ生成手段10に出力されるとともに、プログラムカウンタの値である2000もトレースデータ生成手段10に出力される。

【0066】リソース値生成手段7は、リソースアクセスフラグ生成手段5からビットパターンを受け取ると、このビットパターンのうちの1となっているビットに対応したリソースの値をリソース保持手段3から取り出し、トレースデータ生成手段10に出力する。たとえば、図5の1命令目の場合、先に説明したようにリソースアクセスフラグ生成手段5から、図4のR02, R08, IRに対応したビットが1となるビットパターンが出力されるため、リソース値生成手段7は、リソース保持手段3からレジスタ2, 8および命令語の値である444, 666および10を取り出し、トレースデータ生成手段10に出力する。

【0067】メモリ値生成手段8は、メモリアクセスフラグ生成手段6の出力するビットパターンを受け取ると、このビットパターンのうちの1となっているアドレス指定フィールドを調べる。もし、1に設定されているアドレス指定フィールドがある場合、まずリソース保持手段3から命令語を読み出し、対応するアドレス指定フィールドの内容を取り出す。続いて、メモリ値生成手段8は、アドレス指定フィールドの内容をリソース保持手段3の保持するリソースの値を使って展開し、アクセスされたメモリ1のアドレスを求める。そして、求められたアドレスに対応する内容をメモリ1から取り出し、この内容をトレースデータ生成手段10に出力する。たとえば、図5の1命令目の場合、先に示したようにビットAS1に対応したビットを1に設定したビットパターンがメモリアクセスフラグ生成手段6から出力されるため、メモリ値生成手段8は、リソース保持手段3の保持する命令語のアドレス指定フィールド1の部分を取り出し、アクセスされたアドレスを計算する。図5の1命令目の場合、ここで計算されたアドレスが120となるため、メモリ値生成手段8は、アドレスの120を指定して、メモリ1から888の値を取り出し、これをトレースデータ生成手段10に出力する。

【0068】トレースデータ生成手段10は、リソースアクセスフラグ生成手段5、メモリアクセスフラグ生成手段6およびプログラムカウンタ値生成手段9の出力する各ビットパターンを受け取ると、図3のビットパターンフォーマットに従ったビットパターンを構築し、トレースデータとして出力する。次に、トレースデータ生成手段10は、アクセスのあったリソースやメモリがある場合、リソース値生成手段7の出力するリソースの値お

よびメモリ値生成手段8の出力するリソースの値を図3のビットパターンフォーマットの左に指定されているものから順にトレースデータとして出力する。また、トレースデータ生成手段10は、プログラムカウンタ値生成手段9から出力されたビットIL2~IL0に対応するビットパターンがすべて1の場合、プログラムカウンタ値生成手段9から出力されるプログラムカウンタの値もトレースデータとして出力する。

【0069】したがって、最終的にトレースデータ生成手段10から出力されるトレースデータは、たとえば、図5の1命令目の場合、図3のビットパターンフォーマットにおけるビットR02, R08, AS1およびIRが1、ビットIL2~IL0が0, 1および0であることから、まずはじめに、"0010000010000000110010"のビットパターンが出力される。次に、図3のビットパターンフォーマットの左から定義されている順番にしたがい、レジスタ2, レジスタ8, 命令語およびアドレス指定フィールド1で指定されたアドレスにあるメモリの値の順に444, 666, 10および888がトレースデータとして出力される。

【0070】つまり、図5の1命令目の場合、以下のようなトレースデータが生成されることになる。

【0071】

0010000010000000110010,
444,
666,
10,
888

【0072】また、図5の5命令目の場合、アクセスされたリソースは命令語のみである。一方、プログラムカウンタについては1命令前との変位が、2000-1014=986であり、事前にビットパターンと対応づけられた1, 2または4の値でないため、プログラムカウンタ値生成手段9は、ビットIL2~IL0を1, 1および1と設定したビットパターンをトレースデータ生成手段10に出力する。

【0073】この場合、トレースデータ生成手段10は、命令語以外にプログラムカウンタ値生成手段7から出力されたプログラムカウンタの値も、トレースデータの最後に出力する必要がある。つまり、図5の5命令目の場合、トレースデータ生成手段10は、命令語に対応するビットIRを1、プログラムカウンタの変位に対応するビットIL2~IL0を1, 1および1とした、"0000000000000000100111"のビットパターンをトレースデータとして出力した後、リソース値生成手段7から出力される命令語の20を出力し、その後、プログラムカウンタ値生成手段9から出力されるプログラムカウンタの値の2000を出力する必要がある。

【0074】したがって、図5の5命令目の場合、以下

のようなトレースデータが生成されることになる。

【0075】

000000000000000000100111,
20,
2000

【0076】また、トレースデータ生成手段10は、1命令分のトレースデータをすべて出力し終わると、トレースデータ出力完了指示を、前値更新手段11に出力する。

【0077】前値更新手段11は、トレースデータ出力手段10からのトレースデータ出力完了指示を受け取ると、リソース保持手段3の保持するリソースの値を読み出し、前値保持手段4の保持するリソースの値を更新する。これにより、前値保持手段4は、次の命令に対して1命令前のリソースの値を保持することになる。

【0078】以上のようにして図5の5つの命令で生成されるトレースデータを考えると、各命令毎に出力されるトレースデータは図6のようになる。

【0079】図6において、トレースデータの列が出力されるトレースデータに当たる。情報の内容の列は、トレースデータの値がどのリソースの値を示している情報であり、トレースデータに含まれる情報ではない。

【0080】ここで、第1の実施の形態に係るトレースデータ削減方式におけるトレースデータの情報量について考える。

【0081】まず、各リソースがアクセスされたプログラムカウンタの変位を記憶するビットパターンの情報量を、図3のビットパターンフォーマットから22ビットとする。また、それ以外の情報については、従来例と比較するため、従来例でも使用した図7のトレースデータ上の情報量を使う。

【0082】この条件により求めたビット単位の情報量が図6中の情報量の列であり、各欄の値が、個々のビットパターンやリソースのサイズを示している。そして、合計の行に示された値が、5命令分のトレースデータ全体のビット数である。

【0083】図6の情報量の合計から、本実施の形態での5命令分のトレースデータ量は、462ビットとなっており、図15の従来例の627ビットに比べ、トレースデータの情報量を削減できていることになる。

【0084】次に、本発明の第2の実施の形態について図面を参照して詳細に説明する。

【0085】図8を参照すると、本発明の第2の実施の形態に係るトレースデータ削減方式は、キーボード等の入力装置81と、メモリ1およびプロセッサ2を含む情報処理装置82と、記憶装置83と、ディスプレイ等の出力装置84と、ソフトウェアであるトレースプログラム86を記録した記録媒体85とから構成されている。記録媒体85は、磁気ディスク、半導体メモリ、その他の記録媒体であってもよい。

【0086】図9、図10および図11は、第2の実施の形態に係るトレースデータ削減方式をトレースプログラム86で実現した場合の処理を示すフローチャートである。

【0087】トレースプログラム86は、記録媒体85から情報処理装置82を介して記憶装置83に読み込まれ、情報処理装置82の動作を制御する。情報処理装置82は、トレースプログラム86により、以下の処理を実行する。

10 【0088】デバッグ割り込みが発生すると、プロセッサ2は、自分の持つリソースの値を出力する。

【0089】リソース保持手段3は、プロセッサ2から出力されたリソースの値を、リソース保持手段3の持つリソースの値を格納する配列CRBに格納する。

20 【0090】次に、リソースアクセスフラグ生成手段5は、リソースアクセスフラグ格納用変数RAFおよびワーク用変数Nをそれぞれ0に初期化し（ステップS101）、配列要素CRB(N)と前値保持手段4の持つリソースの値を格納する配列要素BRB(N)とが一致するかどうかを判定する（ステップS102）。両者が一致していれば、リソースアクセスフラグ生成手段5は、

30 リソースアクセスフラグ格納用変数RAFの0ビット目から16ビット目までをリソースアクセスフラグ格納用変数RAFの0ビット目から15ビット目と0とをビット連結したものとする（ステップS103）。両者が一致していなければ、リソースアクセスフラグ生成手段5は、リソースアクセスフラグ格納用変数RAFの0ビット目から16ビット目までをリソースアクセスフラグ格納用変数RAFの0ビット目から15ビット目と1とをビット連結したものとする（ステップS104）。次

に、リソースアクセスフラグ生成手段5は、ワーク用変数Nを1つインクリメントし（ステップS105）、ワーク用変数Nが16未満であれば（ステップS106でノー）、ステップS102に制御を戻してステップS102～ステップS106を繰り返し、ワーク用変数Nが16以上であれば（ステップS106でイエス）、処理を終了する。

40 【0091】続いて、メモリアccessフラグ生成手段6は、メモリアccessフラグ格納用変数MAFおよびワーク用変数Nをそれぞれ0に初期化し（ステップS201）、配列CRB内のプログラムカウンタのN番目のアドレスシラブルのメモリアccessを確認する（ステップS202）。メモリアccessがなければ、メモリアccessフラグ生成手段6は、メモリアccessフラグ格納用変数MAFの0ビット目から2ビット目までをメモリアccessフラグ格納用変数MAFの1ビット目と0とをビット連結したものとし（ステップS203）、メモリアccessがあれば、メモリアccessフラグ格納用変数MAFの0ビット目から2ビット目までをメモリアccessフラグ格納用変数MAFの1ビット目と1とをビット連結し

たものとする(ステップS204)。次に、メモリアクセスフラグ生成手段6は、ワーク用変数Nを1つインクリメントし(ステップS205)、ワーク用変数Nが2未満であれば(ステップS206でノー)、ステップS202に制御を戻してステップS202～ステップS206を繰り返し、ワーク用変数Nが2以上であれば(ステップS206でイエス)、処理を終了する。

【0092】次に、リソース値生成手段7は、リソース数格納用変数RSNを0に、テンポラリ変数TRAFをリソースアクセスフラグ格納用変数RAFに、ワーク用変数Nを0にそれぞれ初期化し(ステップS301)、テンポラリ変数TRAFの0ビット目から1ビット目までが0であるかどうかを判定する(ステップS302)。0でなければ、リソース値生成手段7は、リソースの値を保持する配列要素RVT(RSN)に配列要素CRB(N)を代入し、ソース数格納用変数RSNを1つインクリメントする(ステップS303)。次に、リソース値生成手段7は、テンポラリ変数TRAFの0ビット目から16ビット目をテンポラリ変数TRAFの0ビット目から15ビット目と0とをビット結合したものとし、ワーク用変数Nを1つインクリメントする(ステップS304)。次に、リソース値生成手段7は、ワーク用変数Nが16未満であれば(ステップS305でノー)、ステップS302に制御を戻してステップS302～ステップS305を繰り返し、ワーク用変数Nが16以上であれば(ステップS305でイエス)、処理を終了する。

【0093】続いて、メモリ値生成手段8は、アドレスシラブル数格納用変数ASNを0に、テンポラリ変数TMAFをメモリアクセスフラグ格納用変数MAFに、ワーク用変数Nを0にそれぞれ初期化し(ステップS401)、テンポラリ変数TMAFの0ビット目から1ビット目までが0であるかどうかを判定する(ステップS402)。0でなければ、メモリ値生成手段8は、ASV=CRBに含まれる命令語のNの値に対応したアドレスシラブルを展開したアドレス値とし(ステップS403)、アクセスされたメモリの値を保持する配列要素MVT(ASN)にメモリ(ASV)の値を代入し、アドレスシラブル数ASNを1つインクリメントする(ステップS404)。次に、メモリ値生成手段8は、テンポラリ変数TMAFの0ビット目から2ビット目を、テンポラリ変数TMAFの1ビット目と0とをビット結合したものとし、ワーク用変数Nを1つインクリメントする(ステップS405)。続いて、メモリ値生成手段8は、ワーク用変数Nが2未満であれば(ステップS406でノー)、ステップS402に制御を戻してステップS402～ステップS406を繰り返し、ワーク用変数Nが2以上であれば(ステップS406でイエス)、処理を終了する。

【0094】次に、プログラムカウンタ値生成手段9

は、プログラムカウンタ差分格納用変数PCDを配列CRB内のプログラムカウンタの値からリソース値格納用配列BRB内のプログラムカウンタの値を引いたものとし(ステップS501)、プログラムカウンタの差分PCDが1, 2または4であるかどうかを判定する(ステップS502)。そうでなければ、プログラムカウンタ値生成手段9は、ビットパターン格納用変数ILを7とし、プログラムカウンタ値保持変数PCTをリソース値格納配列CRB内のプログラムカウンタの値とし(ステップS503)、そうであれば、ビットパターン格納用変数ILをプログラムカウンタ差分格納用変数PCDとし(ステップS504)、処理を終了する。

【0095】続いて、トレースデータ生成手段10は、ビットパターンフィールド値格納用変数BPFをリソースアクセスフラグ格納用変数RAF、メモリアクセスフラグ格納用変数MAFおよびビットパターン格納用変数LNのビット結合とし(ステップS601)、ビットパターンフィールド値格納用変数BPFをトレースデータとして出力し(ステップS602)、ワーク用変数Nを0とする(ステップS603)。次に、トレースデータ生成手段10は、ワーク用変数Nがリソース数格納用変数RSN以上であるかどうかを判定し(ステップS604)、ワーク用変数Nがリソース数格納用変数RSN未満であれば、リソースの値を保持する配列要素RVT(N)をトレースデータとして出力し(ステップS605)、ワーク用変数Nを1つインクリメントして(ステップS606)、ステップS604に制御を戻す。ステップS604で、ワーク用変数Nがリソース数格納用変数RSN以上であれば、トレースデータ生成手段10

は、ワーク用変数Nを0とし(ステップS607)、ワーク用変数Nがアドレスシラブル数格納用変数ASN以上であるかどうかを判定する(ステップS608)。ワーク用変数Nがアドレスシラブル数格納用変数ASN未満であれば、メモリの値を保持する配列要素MVT(N)をトレースデータとして出力し(ステップS609)、ワーク用変数Nを1つインクリメントして(ステップS610)、ステップS608に制御を戻す。ステップS608で、ワーク用変数Nがアドレスシラブル格納用変数ASN以上であれば、トレースデータ生成手段10は、ビットパターン格納用変数ILが7かどうかを判定し(ステップS611)、そうであれば、プログラムカウンタ値保持変数PCTをトレースデータとして出力して(ステップS612)、処理を終了する。

【0096】次に、前値更新手段11は、リソース保持手段3の配列CRBを読み出し、前値保持手段4に出力する。

【0097】最後に、前値保持手段4は、前値更新手段11の出力にて配列BFBを更新する。

【0098】

【発明の効果】第1の効果は、トレースデータのデータ

量を削減できることにある。これにより、従来採取が不可能であった大規模なプログラムのトレースデータが採取可能となり、大規模なプログラムの性能評価を行うことを可能とすることである。その理由は、トレースデータとして採取すべき情報処理装置の持つ情報には、いくつかの特性があり、その特性に着目することにより、採取すべきトレースデータの量を減らすことができるからである。

【0099】たとえば、通常の情報処理装置では、1命令毎にプログラムカウンタ、命令語、レジスタなどの複数のリソースがアクセスされる。一方、トレースデータへの出力が必要となるレジスタなどのすべてのリソースの数は、数十程度の場合が一般である。その場合、リソース毎に識別情報を与えるより、本発明のように各リソースと1対1で対応したビットパターンで表現した方が、同時に複数のリソースに対するアクセスの有無を表現できることから、全体としての情報量を減らすことができる。

【0100】また、プログラムカウンタは、多くの場合、プログラムカウンタ全体に対し、比較的小さな数種類の値でインクリメントされるだけである。したがって、プログラムカウンタ全体を1命令毎にトレースデータとして記録することは無駄であり、本発明のように通常は変位に対応した少ないビット数をトレースデータに記録し、このビット幅で表現できなくなったときのみ、プログラムカウンタ全体をトレースデータに記録することにより、トレースデータの情報量を減らすことができる。

【0101】さらに、ある命令がどのアドレスを参照したかは、命令語とアドレスを指定するために使用されるレジスタの値とが既知であれば、計算により求めることが可能であり、トレースデータとして、アクセスされたメモリのアドレスを記録することも無駄である。つまり、本発明のようにメモリをアクセスした命令語内のアドレス指定フィールドを識別する手段のみを提供すれば、多くのビット数を必要とするアドレスの値をトレースデータ内に保持する必要はなく、トレースデータの情報量を減らすことができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態に係るトレースデータ削減方式の構成を示すブロック図である。

【図2】第1の実施の形態を説明するための情報処理装置の条件を示す図である。

【図3】第1の実施の形態で使用するビットパターンのフォーマットを示す図である。

【図4】図3のビットパターンフォーマットでのビット

パターン例を示す図である。

【図5】第1の実施の形態で使用する命令実行イメージを示す図である。

【図6】図5の命令実行イメージを第1の実施の形態を使って採取したトレースデータ列を示した図である。

【図7】第1の実施の形態で採取されるトレースデータ上の各リソースの情報量を示す図である。

【図8】本発明の第2の実施の形態に係るトレースデータ削減方式の構成を示すブロック図である。

10 【図9】第2の実施の形態に係るトレースデータ削減方式の処理の前部を示すフローチャートである。

【図10】第2の実施の形態に係るトレースデータ削減方式の処理の中程部を示すフローチャートである。

【図11】第2の実施の形態に係るトレースデータ削減方式の処理の後部を示すフローチャートである。

【図12】従来例における1つのリソースに対する単位情報の例を示す図である。

【図13】図12の単位情報に対する設定の例を示す図である。

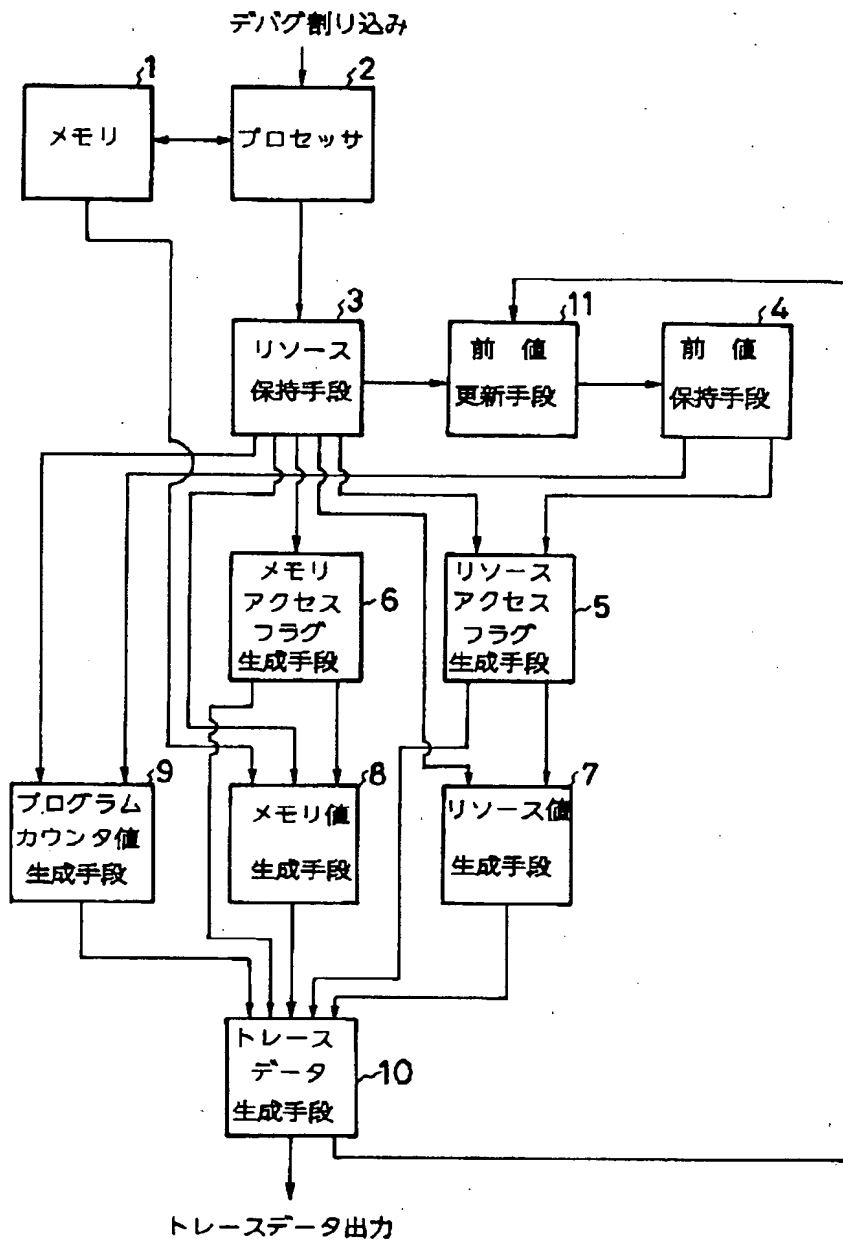
20 【図14】従来例で使用するリソースと識別情報との対応表である。

【図15】従来例で図5の命令実行イメージを使って採取したトレースデータイメージを示す図である。

【符号の説明】

- 1 メモリ
- 2 プロセッサ
- 3 リソース保持手段
- 4 前値保持手段
- 5 リソースアクセスフラグ生成手段
- 30 6 メモリアクセスフラグ生成手段
- 7 リソース値生成手段
- 8 メモリ値生成手段
- 9 プログラムカウンタ値生成手段
- 10 トレースデータ生成手段
- 11 前値更新手段
- 21 識別情報
- 22 位置情報
- 23 内容
- 81 入力装置
- 40 82 情報処理装置
- 83 記憶装置
- 84 出力装置
- 85 記録媒体
- 86 トレースプログラム

【図 1】



【図 3】

ビットパターンフォーマット



【図 4】

【図 2】

情報処理装置条件

レジスタ数	16
アドレス指定フィールド数	2
プログラムカウンタ初期値	1000

【図 7】

トレースデータ上の情報量

情報量	ビット数
識別情報	3
位置情報	レジスタ 4
	メモリ 32
内容	命令語 命令長
	プログラムカウンタ 32
	レジスタ 32
	メモリ 32

【図 14】

識別情報対応表

命令語	I
プログラムカウンタ	P
レジスタ	R
メモリ	M
区切り情報	S

【図 12】

【図 13】



単位情報

識別情報	位置情報	内容
21	22	23

単位情報の例

R	1	1234
21	22	23

【図5】

命令実行イメージ

1 命令目			
リソース	位 置	フィールド位置	アクセス内容
命令語	—	—	1 0
命令語長	—	—	1 6
プログラムカウンタ	—	—	1 0 0 2
レジスタ	2	—	4 4 4
レジスタ	8	—	6 6 6
メモリ	1 2 0	1	8 8 8

2 命令目			
リソース	位 置	フィールド位置	アクセス内容
命令語	—	—	4 5 6
命令語長	—	—	3 2
プログラムカウンタ	—	—	1 0 0 6
レジスタ	1 5	—	1 3 5
メモリ	4 8 0	2	1 0 1 0

3 命令目			
リソース	位 置	フィールド位置	アクセス内容
命令語	—	—	7 8 9
命令語長	—	—	3 2
プログラムカウンタ	—	—	1 0 1 0
レジスタ	7	—	1 2 4

4 命令目			
リソース	位 置	フィールド位置	アクセス内容
命令語	—	—	7 8 9
命令語長	—	—	3 2
プログラムカウンタ	—	—	1 0 1 4
レジスタ	7	—	2 4 8

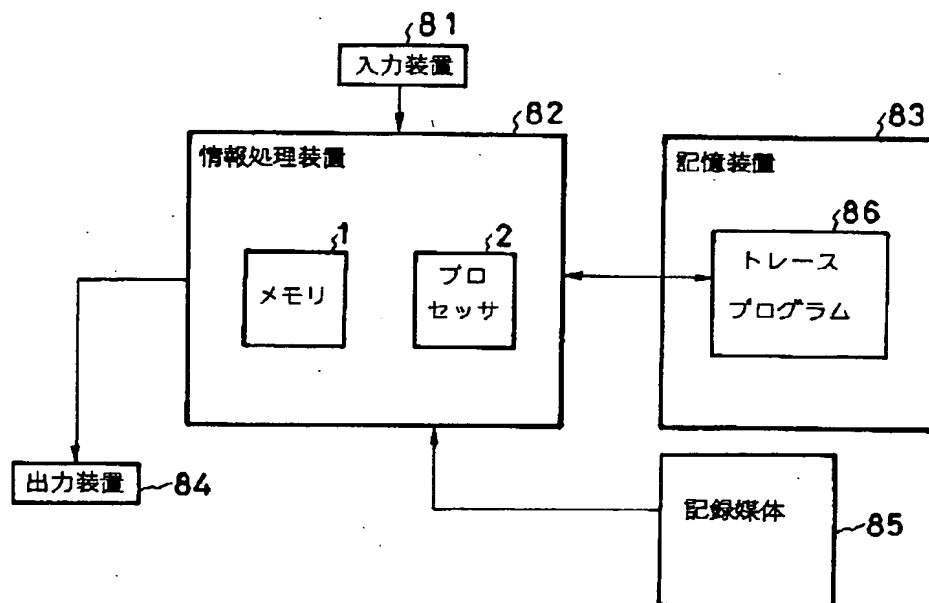
5 命令目			
リソース	位 置	フィールド位置	アクセス内容
命令語	—	—	2 0
命令語長	—	—	1 6
プログラムカウンタ	—	—	2 0 0 0

【図6】

トレースデータイメージ

		トレースデータ	情報の内容	情報量 (bit)
1 命令目	ビットパターン	0010000010000000100000		22
	データ	4 4 4	レジスタ2	32
	データ	6 6 6	レジスタ8	32
	データ	8 8 8	メモリ(フィールド1)	32
	データ	1 0	命令語	16
2 命令目	ビットパターン	000000000000001101100		22
	データ	1 3 5	レジスタ15	32
	データ	8 8 8	メモリ(フィールド2)	32
	データ	4 5 6	命令語	32
3 命令目	ビットパターン	0000000100000000100010		22
	データ	1 2 4	レジスタ7	32
	データ	7 8 9	命令語	32
4 命令目	ビットパターン	0000000100000000000010		22
	データ	4 4 4	レジスタ7	32
5 命令目	ビットパターン	0000000000000000100111		22
	データ	2 0	命令語	16
	データ	2 0 0 0	プログラムカウンタ	32
合計				462

【図8】

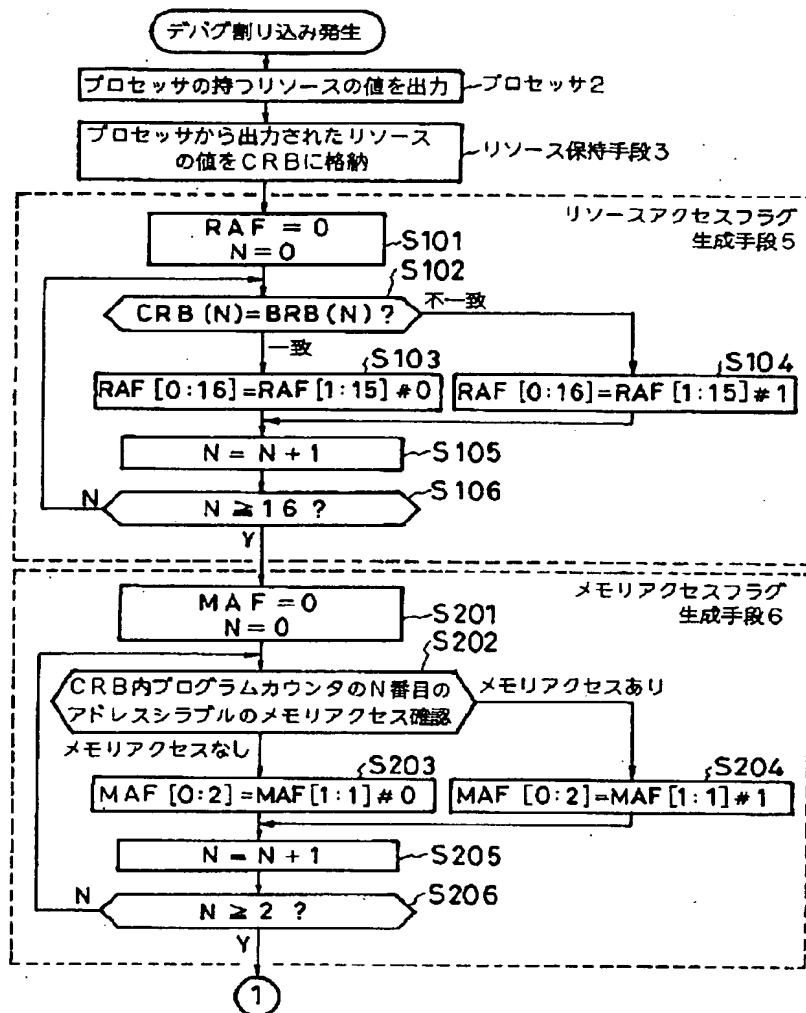


【図15】

トレースデータイメージ

トレースデータ				
	識別情報	位置情報	内 容	情報量 (bit)
1 命令目	S	—	—	3
	P	—	1 0 0 2	35
	I	—	1 0	19
	R	2	4 4 4	39
	R	8	6 6 6	39
	M	1 2 0	8 8 8	67
2 命令目	S	—	—	3
	P	—	1 0 0 6	35
	I	—	4 5 6	35
	R	1 5	1 3 5	39
	M	4 8 0	1 0 1 0	67
3 命令目	S	—	—	3
	P	—	1 0 1 0	35
	I	—	7 8 9	35
	R	7	1 2 4	39
4 命令目	S	—	—	3
	P	—	1 0 1 4	35
	R	7	2 4 8	39
5 命令目	S	—	—	3
	P	—	2 0 0 0	35
	I	—	2 0	19
合計				627

【図9】



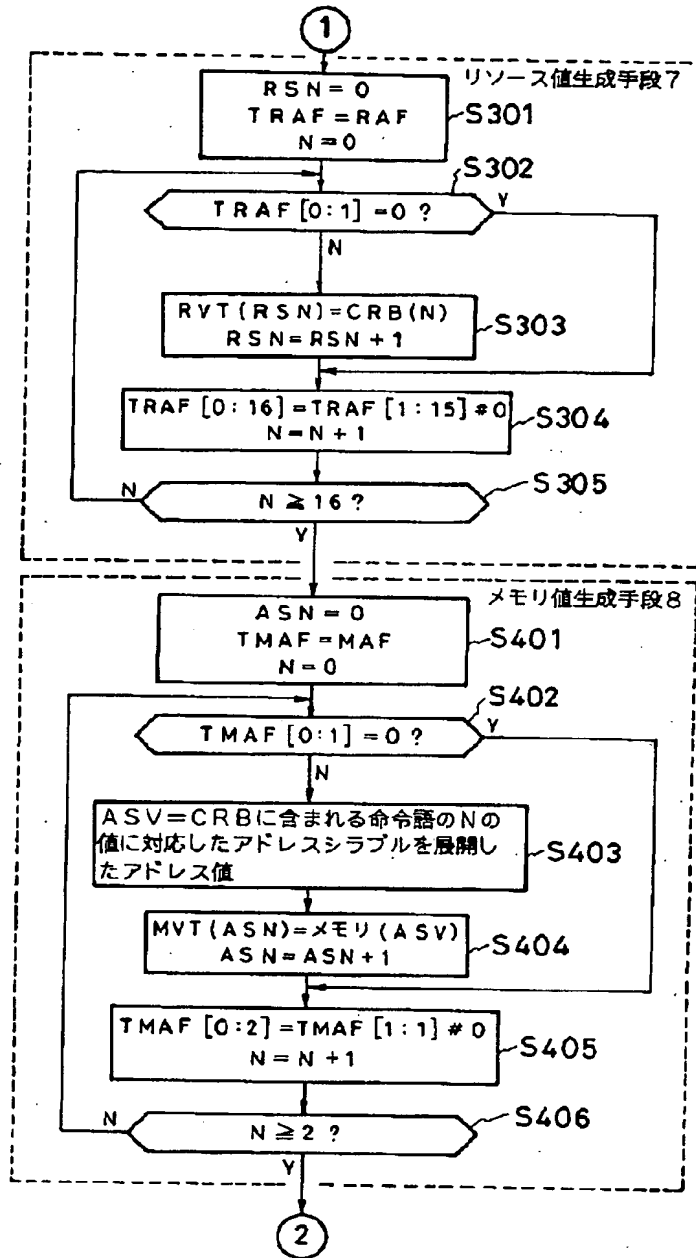
$x(y)$: x の配列の y 番目の位置を指す。

$x[n:m]$: x の変数の n ビット目から m ビット分を示す。 n は0オリジン、 p ビットの変数の場合、上位ビット側を0、下位ビット側を $p-1$ とする。

: ビット連結を示す。 $x = y \# z$ と記述した場合、 $y = 101$ 、 $z = 110$ とすると、 z は101110となる。

メモリ(x) : メモリのアドレス x を示す。

【図10】



【图 1-1】

